

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing Of Claims:**

1. **(Currently Amended)** Method of selecting a proxy cache, said method comprising the steps of:

defining a plurality of proxy caches into which a URL may be hashed;

identifying a candidate set of proxy caches for a given URL based on information related to said URL; and

selecting a proxy cache from the candidate set at least on the basis of latency.

A  
2. **(Original)** The method according to Claim 1, wherein said selecting step comprises selecting a proxy cache from the candidate set at least on the basis of latency and load condition.

3. **(Original)** The method according to Claim 1, wherein said identifying step comprises:

hashing the URL into an anchor hash partition;

forming a candidate set of hash partitions by including one or more nearby partitions into said anchor hash partition; and

mapping each partition to a proxy cache.

4. (Original) The method according to Claim 1, wherein said identifying step comprises:

creating an indirect mapping of hash partitions to a proxy ID array;

hashing the URL into an anchor hash partition;

finding the corresponding anchor proxy cache; and

forming a candidate set of proxy caches by including one or more nearby proxy caches from the proxy ID array into the anchor proxy cache.

5. (Original) The method according to Claim 1, wherein said selecting step comprises selecting a proxy cache from the candidate set of proxies based at least on minimum response time.

6. (Original) The method according to Claim 1, wherein said selecting step comprises selecting a proxy cache from the candidate set of proxies based at least on discounted response time that prefers the anchor proxy cache unless the response time is better by a predetermined amount.

7. (Original) The method according to Claim 1, wherein said selecting step comprises selecting a proxy cache from the candidate set of proxies based at least on the condition that a proxy cache server is not overloaded.

8. (Original) The method according to Claim 3, wherein the nearby partitions comprise partitions with hash values greater than that of the anchor partition.

9. (Original) The method according to Claim 3, wherein the nearby partitions comprise partitions with hash values less than that of the anchor partition.

10. (Original) The method according to Claim 3, wherein the nearby partitions comprise partitions with hash values both greater and less than that of the anchor partition.

11. (Original) The method according to Claim 3, wherein said mapping step comprises hashing each hash partition into a number between 1 and  $P$ , wherein  $P$  represents the total number of proxies.

12. (Original) The method according to Claim 3, wherein said mapping step further comprises:

for each proxy, generating  $N/P$  random numbers between 0 and 1, wherein  $N$  represents the total number of hash partitions and  $P$  represents the total number of proxies;

generating a proxy list by sorting the corresponding  $N$  random numbers generated; and

assigning each hash partition to one proxy based on the sorted proxy list.

13. (Original) The method according to Claim 4, wherein the step of creating an

indirect mapping of hash partitions to a proxy ID array further comprises:

forming a proxy ID array with collaborative proxy caches;

creating a hash partition segment that maps each hash partition to the index of the proxy ID array; and

replicating the hash partition segment for a predetermined number of times.

14. (Currently Amended) System for selecting a proxy cache, said system comprising:

defining a plurality of proxy caches into which a URL may be hashed;

an identifier for identifying a candidate set of proxy caches for a given URL based on information related to said URL; and

a selector for selecting a proxy cache from the candidate set at least on the basis of latency.

15. (Original) The system according to Claim 14, wherein said selector is adapted to select a proxy cache from the candidate set at least on the basis of latency and load condition.

16. (Original) The system according to Claim 14, wherein said identifier is adapted to:

hash the URL into an anchor hash partition;

form a candidate set of hash partitions by including one or more nearby partitions into said anchor hash partition; and  
map each partition to a proxy cache.

17. (Original) The system according to Claim 14, wherein said identifier is adapted to:

create an indirect mapping of hash partitions to a proxy ID array;  
hash the URL into an anchor hash partition;  
find the corresponding anchor proxy cache; and  
form a candidate set of proxy caches by including one or more nearby proxy caches from the proxy ID array into the anchor proxy cache.

18. (Original) The system according to Claim 14, wherein said selector is adapted to select a proxy cache from the candidate set of proxies based at least on minimum response time.

19. (Original) The system according to Claim 14, wherein said selector is adapted to select a proxy cache from the candidate set of proxies based at least on discounted response time that prefers the anchor proxy cache unless the response time is better by a predetermined amount.

20. (Original) The system according to Claim 14, wherein said selector is adapted to select a proxy cache from the candidate set of proxies based at least on the condition that a proxy cache server is not overloaded.

21. (Original) The system according to Claim 16, wherein the nearby partitions comprise partitions with hash values greater than that of the anchor partition.

22. (Original) The system according to Claim 16, wherein the nearby partitions comprise partitions with hash values less than that of the anchor partition.

23. (Original) The system according to Claim 16, wherein the nearby partitions comprise partitions with hash values both greater and less than that of the anchor partition.

24. (Original) The system according to Claim 16, wherein said identifier is adapted to map each hash partition into a number between 1 and  $P$ , wherein  $P$  represents the total number of proxies.

25. (Original) The system according to Claim 16, wherein said identifier, in mapping, is adapted to:

for each proxy, generate  $N/P$  random numbers between 0 and 1, wherein  $N$  represents the total number of hash partitions and  $P$  represents the total number of proxies;

generate a proxy list by sorting the corresponding  $N$  random numbers generated; and

assign each hash partition to one proxy based on the sorted proxy list.

26. (Original) The system according to Claim 17, wherein said identifier, in creating an indirect mapping, of hash partitions to a proxy ID array, is adapted to:

form a proxy ID array with collaborative proxy caches;

create a hash partition segment that maps each hash partition to the index of the proxy ID array; and

replicate the hash partition segment for a predetermined number of times.

27. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for selecting a proxy cache, said method steps comprising:

defining a plurality of proxy caches into which a URL may be hashed;

identifying a candidate set of proxy caches for a given URL based on information related to said URL; and

selecting a proxy cache from the candidate set at least on the basis of latency.